

PATENT
ROC920010202US1

P R E L I M I N A R Y A M E N D M E N T

IN THE CLAIMS:

Please **amend** Claims 5, 6, 7, 9, 17, 19 and 22; **add** Claim 25; and **cancel** Claims 1 to 4, 8, 13 to 16, 20, 21, 23, and 24, so that the claims read as follows:

1. to 4. (Canceled)

5. (Currently Amended) A method of dequeuing a flow from a scheduling queue having an empty indicator, the method comprising:

~~examining an empty indicator associated with the scheduling queue;~~

determining if an empty indicator of a scheduling queue is set to empty;

~~refraining from searching the scheduling queue if the empty indicator indicates that the scheduling queue is empty;~~

searching the scheduling queue if the empty indicator indicates that the scheduling queue is not empty a flow is associated with the scheduling queue;

determining if the scheduling queue is empty based on the search;

setting the empty indicator to empty if the search determines that the scheduling queue is empty;
and

~~detaching from the scheduling queue a winning flow found in the searching step~~ the flow associated with the scheduling queue found when the scheduling queue is searched.

PATENT
ROC920010202US1

6. (Currently Amended) The method of claim 5, further comprising, ~~prior to the examining step,~~ selecting the scheduling queue from among a plurality of scheduling queues in a round robin process.

7. (Original) The method of claim 5, wherein the searching step includes searching a plurality of subqueues included in the scheduling queue, the subqueues having mutually different respective ranges and resolutions.

8. (Canceled)

9. (Currently Amended) The method as recited in claim 5, further ~~A method of enqueueing a flow to a scheduling queue,~~ comprising:

attaching a flow to the scheduling queue;

and

placing the ~~an~~ empty indicator ~~of a~~ plurality of ~~empty indicators~~ associated with the scheduling queue in a condition to indicate that the scheduling queue is not empty.

10. (Original) The method of claim 9, wherein the attaching step includes assigning the flow to a slot in the scheduling queue according to the formula $CP + ((WF \times FS) / SF)$, where:

CP is a pointer that indicates a current position in the scheduling queue;

WF is a weighting factor associated with the flow;

PATENT
ROC920010202US1

FS is a size of a data frame associated with the flow; and

SF is a scaling factor.

11. (Original) The method of claim 9, wherein the placing step includes setting a bit in a register.

12. (Original) The method of claim 9, wherein the placing step includes resetting a bit in a register.

13. to 16. (Canceled)

17. (Currently Amended) The method of claim 5 13, wherein, if the detaching step is performed, a further search of the scheduling queue is performed to determine whether any flows are enqueued in the scheduling queue other than the flow detached in the detaching step.

18. (Original) The method of claim 17, wherein the empty indicator is placed in a condition to indicate that the scheduling queue is empty if the further search of the scheduling queue determines that there are no flows in the scheduling queue other than the flow detached in the detaching step.

19. (Currently Amended) A scheduler for a network processor, comprising:

PATENT
ROC920010202US1

one or more scheduling queues, each adapted to define a respective sequence in which flows are to be serviced; and

a plurality of empty indicators, each empty indicator of the plurality of empty indicators being associated with a respective scheduling queue to indicate whether the respective scheduling queue is empty;

wherein the scheduler is adapted to:

determine if an empty indicator of the plurality of empty indicators is set to empty;

search the scheduling queue if the empty indicator indicates that a flow is associated with the scheduling queue;

determine if the scheduling queue is empty based on the search;

set the empty indicator to empty if the search determines that the scheduling queue is empty; and

detach the flow associated with the scheduling queue wherein the flow is found when the scheduling queue is searched

~~examine an empty indicator associated with a first scheduling queue;~~

~~refrain from searching the first scheduling queue if the empty indicator indicates that the first scheduling queue is empty;~~

~~search the first scheduling queue if the empty indicator indicates that the first scheduling queue is not empty; and~~

PATENT
ROC920010202US1

~~detach from the first scheduling queue a winning flow found in the search of the first scheduling queue.~~

20. (Canceled)

21. (Canceled)

22. (Currently Amended) A computer program product adapted to dequeue a flow from a scheduling queue, the computer program product comprising:

a medium readable by a computer, the computer readable medium having computer program code adapted to:

determine if an empty indicator is set to empty;

search the scheduling queue if the empty indicator indicates that a flow is associated with the scheduling queue;

determine if the scheduling queue is empty based on the search;

set the empty indicator to empty if the search determines that the scheduling queue is empty; and

detach the flow associated with the scheduling queue wherein the flow is found when the scheduling queue is searched

~~examine an empty indicator of a plurality of empty indicators associated with the scheduling queue;~~

PATENT
ROC920010202US1

~~refrain from searching the
scheduling queue if the empty indicator indicates that
the scheduling queue is empty;~~

~~search the scheduling queue if the
empty indicator indicates that the scheduling queue is
not empty; and~~

~~detach from the scheduling queue a
winning flow found in the search of the scheduling
queue.~~

23. (Canceled)

24. (Canceled)

25. (New) A method of enqueueing a flow to a
scheduling queue, comprising:

attaching a flow to the scheduling queue;

and

placing an empty indicator of a plurality of
empty indicators associated with the scheduling queue
in a condition to indicate that the scheduling queue
is not empty;

wherein the attaching step includes
assigning the flow to a slot in the scheduling queue
according to the formula $CP + ((WF \times FS) / SF)$, where:

CP is a pointer that indicates a current
position in the scheduling queue;

WF is a weighting factor associated with the
flow;

FS is a size of a data frame associated with
the flow; and

SF is a scaling factor.